

SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA

```

PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      EEEEEEEEEEE
PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      EEEEEEEEEEE
PP           PP  AA          AA  RR          RR  SS      EE
PP           PP  AA          AA  RR          RR  SS      EE
PP           PP  AA          AA  RR          RR  SS      EE
PP           PP  AA          AA  RR          RR  SS      EE
PPPPPPPP      AA          AA  RRRRRRRR      SSSSSS      EEEEEEEEE
PPPPPPPP      AA          AA  RRRRRRRR      SSSSSS      EEEEEEEEE
PP           AAAAAAAAAA  RR  RR          SS      EE
PP           AAAAAAAAAA  RR  RR          SS      EE
PP           AA          AA  RR          RR          SS      EE
PP           AA          AA  RR          RR          SS      EE
PP           AA          AA  RR          RR          SS      EE
PP           AA          AA  RR          RR          SSSSSSSS      EEEEEEEEEEE
PP           AA          AA  RR          RR          SSSSSSSS      EEEEEEEEEEE

```

```

LL               IIIIII             SSSSSSSS
LL               IIIIII             SSSSSSSS
                II                  SS
LL              II                  SS
LL              II                  SS
LL              II                  SS
LL              II                  SS
LL              II                  SSSSSS
LL              II                  SSSSSS
LL              II                  SS
LL              II                  SS
LL              II                  SS
LL              II                  SS
LLLLLLLLLLLL    IIIIII             SSSSSSSS
LLLLLLLLLLLL    IIIIII             SSSSSSSS

```

(1)	2	COPYRIGHT NOTICE
(1)	29	PROGRAM DESCRIPTION
(2)	107	DECLARATIONS
(3)	118	DATA STORAGE
(4)	134	PARSER STATE TABLE
(5)	991	EXPRESSION ANALYSIS ACTION ROUTINES
(6)	1094	STORE_RELOCATION, STORE READ/RELOCATE VALUE
(6)	1114	STORE_PROC_INDEX, STORE PROCESS INDEX LOW BITS
(7)	1137	SAVE_COMMAND, SAVE THE CURRENT COMMAND
(8)	1172	REPEAT_COMMAND, REPEAT THE PREVIOUS COMMAND
(9)	1194	DEFINE_KEY - DEFINE KEY
(10)	1214	Action Routines for SET:SHOW RMS/DISPLAY=optionspec

```

0000 1 .TITLE PARSE Parse tables for SDA commands
0000 2 .SBTTL COPYRIGHT NOTICE
0000 3 .IDENT 'V04-000'
0000 4 :
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :

```

```

0000 29 .SBTTL PROGRAM DESCRIPTION
0000 30 :++
0000 31 FACILITY
0000 32
0000 33 SYSTEM DUMP ANALYZER
0000 34
0000 35 ABSTRACT
0000 36
0000 37 This module contains the parse tables which
0000 38 are used to parse SDA commands.
0000 39
0000 40 ENVIRONMENT
0000 41
0000 42 NATIVE MODE, USER MODE
0000 43
0000 44 AUTHOR
0000 45
0000 46 TIM HALVORSEN, JULY 1978
0000 47
0000 48 MODIFIED BY
0000 49
0000 50 V03-015 EMB0109 Ellen M. Batbouta 24-Jul-1984
0000 51 Fix SHOW DEVICE/ADDRESS so that it will be consistent
0000 52 with other sda commands with the qualifier /address
0000 53 such as SHOW PORTS/ADDRESS.
0000 54
0000 55 V03-014 EMB0102 Ellen M. Batbouta 30-May-1984
0000 56 Add support for SHOW CLUSTER, SHOW CONNECTIONS, SHOW
0000 57 RSPID, and SHOW PORTS.
0000 58
0000 59 V03-013 PRB0302 Paul Beck 28-Dec-1983 17:50
0000 60 Add support for EXAMINE/INST/NOSKIP
0000 61
0000 62 V03-012 ROW0260 Ralph O. Weber 21-NOV-1983
0000 63 Changes parse tables to allow SHOW RESOURCES, SHOW LOCKS, and
0000 64 SHOW PROCESS /LOCKS.
0000 65
0000 66 V03-011 ROW0237 Ralph O. Weber 10-NOV-1983
0000 67 Add TPARSE table entries to support the SHOW DEVICE/ADDR
0000 68 command.
0000 69
0000 70 V03-010 WMC0002 Wayne Cardoza 30-Sep-1983
0000 71 Bad transition after page table range
0000 72
0000 73 V03-009 BLS0230 Benn Schreiber 24-Jul-1983
0000 74 Correct end of line / comment handling
0000 75
0000 76 V03-008 WMC0001 Wayne cardoza 11-Jul-1983
0000 77 Support for separate display of P0 and P1 process page tables.
0000 78
0000 79 V03-007 JLV0259 Jake VanNoy 23-MAY-1983
0000 80 ADD DEFINE/KEY.
0000 81
0000 82 V03-006 TMK0001 Todd M. Katz 21-Mar-1983
0000 83 ADD SET LOG and SET NOLOG syntax.
0000 84
0000 85 V03-005 CWH1002 CW Hobbs 2-Mar-1983

```

0000	86	:			
0000	87	:			
0000	88	:			
0000	89	:			
0000	90	:			
0000	91	:			
0000	92	:	V03-004	RAS0123	Ron Schaefer
0000	93	:			8-Feb-1983
0000	94	:			
0000	95	:	V03-003	JLV0224	Jake VanNoy
0000	96	:			21-JAN-1983
0000	97	:			
0000	98	:			
0000	99	:	V03-002	LMP0031	L. Mark Pilant,
0000	100	:			17-Jun-1982 13:15
0000	101	:			
0000	102	:	V03-001	MSH0001	Maryann Hinden
0000	103	:			10-Jun-1982
0000	104	:			
0000	105	:--			

Store only the low bits of proc index (as determined by the global PIX WIDTH so that we can interchange internal and extended pids. Change syntax of SHOW SUMMARY to add SHOW SUMMARY /IMAGE.

Add SHOW PROC/RMS=RJB syntax.

Add Examine/PSL Examine/CONDITION_VLAUE and Examine/TIME parsing. Add VALIDATE QUEUE syntax.

Add support for SHOW PROCESS/CHANNELS. Also, change the parsing to allow a process name along with qualifiers.

Add SHOW LOCK, SHOW RES, SHOW PROC/LOCK commands.

	0000	107		.SBTTL	DECLARATIONS	
	0000	108	:			
	0000	109	:	SYMBOL	DEFINTIONS	
	0000	110	:			
	0000	111		\$TPADEF		; TPARSE DEFINITIONS
	0000	112		\$OPTDEF		; DEFINE BITS IN OPTIONS WORD
	0000	113		\$SMGDEF		; SCREEN INPUT DEFINITIONS
	0000	114				
0000003B	0000	115	SEMI	=	^A':'	
0000003A	0000	116	COLON	=	^A':'	

	0000	118	.SBTTL DATA STORAGE	
	0000	119	:	
	0000	120	:	STORAGE DEFINITIONS
	0000	121	:	
	0000	122	:	
00000000	0000	123	.PSECT SDADATA,WRT,NOEXE	
	0000	124		
00000050	0000	125	PREV_BUFFER:	
	0050	126	.BLKB 80	; HOLDS PREVIOUS COMMAND
00000000	0050	127	PREV_COMMAND:	
00000000	0054	128	.LONG 0	
	0058	129	.ADDRESS PREV_BUFFER	
	0058	130		
00000002	0058	131	KEY_ATTR:	
	0058	132	.LONG SMG\$M_KEY_TERMINATE	; KEYPAD ATTRIBUTE FOR 'REPEAT KEY'

```

005C 134 .SBTTL  PARSE STATE TABLE
005C 135 :
005C 136 :
005C 137 :
005C 138 :
005C 139 $INIT_STATE SDA_STATE,SDA_KEY
005C 140
005C 141 $STATE  START
005C 142 $STRAN  'READ',READ_CMD
005C 143 $STRAN  'REPEAT',START,REPEAT_COMMAND
005C 144 $STRAN  TPAS_LAMBDA,,SAVE_COMMAND
005C 145 $STATE
005C 146 $STRAN  '@',INDIRECT_CMD
005C 147 $STRAN  'COPY',COPY_CMD
005C 148 $STRAN  'DEFINE',DEFINE
005C 149 $STRAN  'EXAMINE',EXAMINE
005C 150 $STRAN  'FORMAT',FORMAT_CMD
005C 151 $STRAN  'HELP',HELP_CMD
005C 152 $STRAN  'SET',SET
005C 153 $STRAN  'SEARCH',SEARCH
005C 154 $STRAN  'SHOW',SHOW
005C 155 $STRAN  'EVALUATE',EVALUATE
005C 156 $STRAN  'EXIT',EXIT_CMD
005C 157 $STRAN  'VALIDATE',VALIDATE
005C 158 $STRAN  TPAS_LAMBDA,CK_EOS
005C 159
005C 160 $STATE  CK_EOS
005C 161 $STRAN  '!$',TPAS_EXIT
005C 162 $STRAN  TPAS_EOS,TPAS_EXIT
005C 163
005C 164 :
005C 165 :
005C 166 :
005C 167 :
005C 168
005C 169 $STATE  SEARCH
005C 170 $STRAN  !EXPRESSION
005C 171 $STATE
005C 172 $STRAN  SEMI,,,OPT$M_LENGTH,OPTIONS
005C 173 $STRAN  COLON,,,OPT$M_RANGE,OPTIONS
005C 174 $STATE
005C 175 $STRAN  !EXPRESSION
005C 176 $STATE
005C 177 $STRAN  '='
005C 178 $STRAN  TPAS_LAMBDA
005C 179 $STATE
005C 180 $STRAN  !EXPRESSION
005C 181 $STATE
005C 182 $STRAN  TPAS_LAMBDA,TPAS_EXIT,SEARCH_MEMORY
005C 183
005C 184 :
005C 185 :
005C 186 :
005C 187
005C 188 $STATE  INDIRECT_CMD
005C 189 $STRAN  !COLLECT,TPAS_EXIT,INDIRECT_COMMAND,,FILE_DESC
005C 190

```

```

005C 191 :
005C 192 : SHOW PROCESS name /options
005C 193 : SHOW SUMMARY
005C 194 : SHOW PAGE TABLE
005C 195 : SHOW PFN DATA
005C 196 : SHOW POOL /IRP /NONPAGED /ALL
005C 197 : SHOW CRASH
005C 198 : SHOW STACK /mode (default is current stack)
005C 199 : SHOW SYMBOL name /ALL
005C 200 : SHOW DEVICE device /ALL
005C 201 : SHOW RMS
005C 202 : SHOW LOCKS
005C 203 : SHOW RESOURCES
005C 204 : SHOW CLUSTER /CSID=n /SCS
005C 205 : SHOW CONNECTIONS /ADDRESS=n
005C 206 : SHOW PORTS /ADDRESS=n
005C 207 : SHOW RSPID /CONNECTION=n
005C 208 :
005C 209 :
005C 210 : $STATE SHOW
005C 211 : $STRAN 'CRASH',TPAS_EXIT,DISPLAY_CRASH
005C 212 : $STRAN 'DEVICE',DEVICE
005C 213 : $STRAN 'HEADER',TPAS_EXIT,PRINT_HEADER
005C 214 : $STRAN 'LOCKS',LOCK
005C 215 : $STRAN 'PAGE TABLE',PAGE_TABLE
005C 216 : $STRAN 'PFN DATA',PFN_DATA
005C 217 : $STRAN 'POOL',POOL
005C 218 : $STRAN 'PROCESS',PROCESS
005C 219 : $STRAN 'RESOURCES',RES
005C 220 : $STRAN 'RMS',SHOW RMS
005C 221 : $STRAN 'STACKS',STACK
005C 222 : $STRAN 'SUMMARY',SHOW SUMMARY
005C 223 : $STRAN 'SYMBOLS',SYMBOL
005C 224 : $STRAN 'CLUSTER',VAXCLUSTER
005C 225 : $STRAN 'CONNECTIONS',SCS_CONNECT
005C 226 : $STRAN 'RSPID',SCS_RDT
005C 227 : $STRAN 'PORTS',SCS_PORTS
005C 228 :
005C 229 : SHOW LOCK lockid
005C 230 : [ /ALL ]
005C 231 :
005C 232 :
005C 233 : $STATE LOCK
005C 234 : $STRAN !LOCK_OPTS,TPAS_EXIT
005C 235 : $STRAN TPAS_REX,,,LOCKID
005C 236 :
005C 237 : $STATE
005C 238 : $STRAN !CK_EOS,TPAS_EXIT,SHOW_ONE_LOCK
005C 239 :
005C 240 : $STATE LOCK_OPTS
005C 241 : $STRAN !CK_EOS,TPAS_EXIT,SHOW_ALL_LOCKS
005C 242 : $STRAN !/
005C 243 :
005C 244 : $STATE
005C 245 : $STRAN 'ALL'
005C 246 :
005C 247 : $STATE

```

005C	248	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_ALL_LOCKS
005C	249		
005C	250		
005C	251	SHOW RES	[/ALL]
005C	252		/LOCKID = xxxx
005C	253		
005C	254		
005C	255	\$STATE	RES
005C	256	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_ALL_RES
005C	257	\$STRAN	!/'
005C	258		
005C	259	\$STATE	
005C	260	\$STRAN	'ALL',TPAS_EXIT,SHOW_ALL_RES
005C	261	\$STRAN	'LOCKID'
005C	262		
005C	263	\$STATE	
005C	264	\$STRAN	':'
005C	265	\$STRAN	'='
005C	266		
005C	267	\$STATE	
005C	268	\$STRAN	TPAS_HEX,,,,LOCKID
005C	269		
005C	270	\$STATE	
005C	271	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_ONE_RES
005C	272		
005C	273		
005C	274	SHOW RMS	
005C	275		
005C	276		
005C	277	\$STATE	SHOW RMS
005C	278	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_RMS_OPT
005C	279		
005C	280	SHOW CLUSTER	/CSID=n /SCS
005C	281		
005C	282	\$STATE	VAXCLUSTER
005C	283	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_CLUSTER
005C	284	\$STRAN	!/'
005C	285		
005C	286	\$STATE	
005C	287	\$STRAN	'CSID'
005C	288	\$STRAN	'SCS',TPAS_EXIT,SHOW_SCS
005C	289		
005C	290	\$STATE	
005C	291	\$STRAN	'='
005C	292		
005C	293	\$STATE	
005C	294	\$STRAN	TPAS_HEX,,,,CSID
005C	295		
005C	296	\$STATE	
005C	297	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_CLUSTER
005C	298		
005C	299		
005C	300	SHOW CONNECTIONS	/ADDR=n
005C	301		
005C	302		
005C	303	\$STATE	SCS_CONNECT
005C	304	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_CONNECTIONS

005C	305	\$STRAN	'/'
005C	306		
005C	307	\$STATE	
005C	308	\$STRAN	'ADDRESS'
005C	309		
005C	310	\$STATE	
005C	311	\$STRAN	'='
005C	312		
005C	313	\$STATE	
005C	314	\$STRAN	!EXPRESSION,TPAS_EXIT,CDT_BYADDR
005C	315	:	
005C	316	:	SHOW PORTS /ADDR=n
005C	317	:	
005C	318		
005C	319	\$STATE	SCS_PORTS
005C	320	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_PORTS
005C	321	\$STRAN	'/'
005C	322		
005C	323	\$STATE	
005C	324	\$STRAN	'ADDRESS'
005C	325		
005C	326	\$STATE	
005C	327	\$STRAN	'='
005C	328		
005C	329	\$STATE	
005C	330	\$STRAN	!EXPRESSION,TPAS_EXIT,PDT_BYADDR
005C	331	:	
005C	332	:	SHOW RSPID /CONNECTION=n
005C	333	:	
005C	334		
005C	335	\$STATE	SCS_RDT
005C	336	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_RSPID
005C	337	\$STRAN	'/'
005C	338		
005C	339	\$STATE	
005C	340	\$STRAN	'CONNECTION'
005C	341		
005C	342	\$STATE	
005C	343	\$STRAN	'='
005C	344		
005C	345	\$STATE	
005C	346	\$STRAN	!EXPRESSION,,,1,CDT_SPCFY
005C	347		
005C	348	\$STATE	
005C	349	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,SHOW_RSPID
005C	350		
005C	351		
005C	352	:	
005C	353	:	SET LOG filespec
005C	354	:	SET NOLOG
005C	355	:	SET OUTPUT filespec
005C	356	:	SET PROCESS name /INDEX=n
005C	357	:	SET RMS/DISPLAY=(opt1,opt2,,,,,optn)
005C	358	:	
005C	359		
005C	360	\$STATE	SET
005C	361	\$STRAN	'LOG',SET_LOG

005C	362	\$STRAN	'NOLOG',TPAS_EXIT,CLOSE_LOG
005C	363	\$STRAN	'OUTPUT',SET_OUTPUT
005C	364	\$STRAN	'PROCESS',SET_PROC
005C	365	\$STRAN	'RMS',SET_RMS
005C	366		
005C	367	:	
005C	368	:	
005C	369	:	
005C	370		SET LOG filespec
005C	371	\$STATE	SET_LOG
005C	372	\$STRAN	!COLLECT,,,,LOG_FILE
005C	373	\$STATE	
005C	374	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,OPEN_LOG
005C	375		
005C	376	:	
005C	377	:	
005C	378	:	
005C	379		SET OUTPUT filespec
005C	380	\$STATE	SET_OUTPUT
005C	381	\$STRAN	!COLLECT,,,,OUTPUT_FILE
005C	382	\$STATE	
005C	383	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,OPEN_OUTPUT
005C	384		
005C	385	:	
005C	386	:	
005C	387	:	
005C	388		SUB-EXPRESSION TO COLLECT ALL REMAINING CHARACTERS
005C	389	\$STATE	COLLECT
005C	390	\$STRAN	TPAS_ANY ; MUST HAVE AT LEAST 1 CHAR.
005C	391	\$STATE	COLLECT_LOOP
005C	392	\$STRAN	!SNARF,COLLECT_LOOP
005C	393	\$STRAN	TPAS_LAMBDA,TPAS_EXIT
005C	394	\$STATE	SNARF
005C	395	\$STRAN	!CK EOS,TPAS_FAIL
005C	396	\$STRAN	TPAS_ANY,TPAS_EXIT
005C	397		
005C	398	:	
005C	399	:	
005C	400	:	
005C	401	:	
005C	402		SET RMS=opt (opt1,opt2,,,,optn)
005C	403	\$STATE	SET_RMS
005C	404	\$STRAN	!DISPLAY,TPAS_EXIT,END_SET_RMS
005C	405		
005C	406	\$STATE	DISPLAY
005C	407	\$STRAN	TPAS_LAMBDA,,SETUP_RMS_TMP
005C	408	\$STATE	
005C	409	\$STRAN	':'
005C	410	\$STRAN	'='
005C	411	\$STATE	
005C	412	\$STRAN	!DIS_OPT,TPAS_EXIT
005C	413	\$STRAN	!('
005C	414	\$STATE	NXT_OPT
005C	415	\$STRAN	!DIS_OPT
005C	416	\$STATE	
005C	417	\$STRAN	<'>,NXT_OPT
005C	418	\$STRAN	'',TPAS_EXIT

005C	419	
005C	420	\$STATE DIS_OPT
005C	421	\$STRAN TPAS_LAMBDA,,SETUP_DIS_OPT
005C	422	\$STATE
005C	423	\$STRAN !ATM_OPT,DIS_OPT_EXIT
005C	424	\$STRAN 'N'
005C	425	\$STATE
005C	426	\$STRAN 'O'
005C	427	\$STATE
005C	428	\$STRAN !ATM_OPT,,OPTSM_NO,RMS_DIS_TMP1
005C	429	\$STATE DIS_OPT_EXIT
005C	430	\$STRAN TPAS_LAMBDA,TPAS_EXIT,STORE_TMP1_OPT
005C	431	
005C	432	\$STATE ATM_OPT
005C	433	\$STRAN 'ALC',IFI,,OPTSM_RMSALL,RMS_DIS_TMP1
005C	434	\$STRAN 'ASB',TPAS_EXIT,,OPTSM_ASB,RMS_DIS_TMP1
005C	435	\$STRAN 'BDB',TPAS_EXIT,,OPTSM_BDB,RMS_DIS_TMP1
005C	436	\$STRAN 'BDBSUM',TPAS_EXIT,,OPTSM_BDBSUM,RMS_DIS_TMP1
005C	437	\$STRAN 'BLB',TPAS_EXIT,,OPTSM_BLB,RMS_DIS_TMP1
005C	438	\$STRAN 'BLBSUM',TPAS_EXIT,,OPTSM_BLBSUM,RMS_DIS_TMP1
005C	439	\$STRAN 'CCB',TPAS_EXIT,,OPTSM_CCB,RMS_DIS_TMP1
005C	440	\$STRAN 'FAB',TPAS_EXIT,,OPTSM_FAB,RMS_DIS_TMP1
005C	441	\$STRAN 'FCB',TPAS_EXIT,,OPTSM_FCB,RMS_DIS_TMP1
005C	442	\$STRAN 'FWA',TPAS_EXIT,,OPTSM_FWA,RMS_DIS_TMP1
005C	443	\$STRAN 'GBD',TPAS_EXIT,,OPTSM_GBD,RMS_DIS_TMP1
005C	444	\$STRAN 'GBDSUM',TPAS_EXIT,,OPTSM_GBDSUM,RMS_DIS_TMP1
005C	445	\$STRAN 'GBH',TPAS_EXIT,,OPTSM_GBH,RMS_DIS_TMP1
005C	446	\$STRAN 'IDX',TPAS_EXIT,,OPTSM_IDX,RMS_DIS_TMP1
005C	447	\$STRAN 'IFAB',IFI,,OPTSM_IFB,RMS_DIS_TMP1
005C	448	\$STRAN 'IFB',IFI,,OPTSM_IFB,RMS_DIS_TMP1
005C	449	\$STRAN 'IRAB',TPAS_EXIT,,OPTSM_IRB,RMS_DIS_TMP1
005C	450	\$STRAN 'IRB',TPAS_EXIT,,OPTSM_IRB,RMS_DIS_TMP1
005C	451	\$STRAN 'NAM',TPAS_EXIT,,OPTSM_NAM,RMS_DIS_TMP1
005C	452	\$STRAN 'RAB',TPAS_EXIT,,OPTSM_RAB,RMS_DIS_TMP1
005C	453	\$STRAN 'RJB',TPAS_EXIT,,OPTSM_RJB,RMS_DIS_TMP1
005C	454	\$STRAN 'RLB',TPAS_EXIT,,OPTSM_RLB,RMS_DIS_TMP1
005C	455	\$STRAN 'TRC',TPAS_EXIT,,OPTSM_TRC,RMS_DIS_TMP1
005C	456	\$STRAN 'WCB',TPAS_EXIT,,OPTSM_WCB,RMS_DIS_TMP1
005C	457	\$STRAN 'XAB',TPAS_EXIT,,OPTSM_XAB,RMS_DIS_TMP1
005C	458	\$STRAN '*',TPAS_EXIT,STORE_STAR
005C	459	
005C	460	\$STATE IFI
005C	461	\$STRAN ':'
005C	462	\$STRAN '='
005C	463	\$STRAN TPAS_LAMBDA,TPAS_EXIT
005C	464	\$STATE IFI2
005C	465	\$STRAN 'ALL',TPAS_EXIT,CLR_IFI_TMP
005C	466	\$STRAN !EXPRESSION,TPAS_EXIT,STORE_IFI_TMP
005C	467	
005C	468	:
005C	469	:
005C	470	:
005C	471	
005C	472	\$STATE EVALUATE
005C	473	\$STRAN '/'
005C	474	\$STRAN TPAS_LAMBDA,EVAL_EXP
005C	475	\$STATE

EVALUATE expression

005C	476	\$STRAN	'CONDITION_VALUE',...OPTSM_COND,OPTIONS
005C	477	\$STATE	EVAL_EXP
005C	478	\$STRAN	!EXPRESSION
005C	479	\$STATE	
005C	480	\$STRAN	'/' EVAL_QUAL
005C	481	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,SHOW_EXPR
005C	482	\$STATE	EVAL_QUAL
005C	483	\$STRAN	'CONDITION_VALUE',...OPTSM_COND,OPTIONS
005C	484	\$STATE	
005C	485	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,SHOW_EXPR
005C	486		
005C	487		
005C	488		
005C	489		
005C	490		
005C	491		
005C	492	\$STATE	EXAMINE
005C	493	\$STRAN	'/'
005C	494	\$STRAN	TPAS_LAMBDA,EXAMINE_EXPRESSION
005C	495	\$STATE	
005C	496	\$STRAN	'P0',EXAM_OK,,OPTSM_P0,OPTIONS
005C	497	\$STRAN	'P1',EXAM_OK,,OPTSM_P1,OPTIONS
005C	498	\$STRAN	'SYSTEM',EXAM_OK,,OPTSM_SYSTEM,OPTIONS
005C	499	\$STRAN	'ALL',EXAM_OK,,OPTSM_ALL,OPTIONS
005C	500	\$STRAN	'INSTRUCTION',EXAM_INS,,OPTSM_INST,OPTIONS
005C	501	\$STRAN	'PSL',,,OPTSM_PSL,OPTIONS
005C	502	\$STRAN	'TIME',,,OPTSM_TIME,OPTIONS
005C	503	\$STRAN	'CONDITION_VALUE',...OPTSM_COND,OPTIONS
005C	504		
005C	505	\$STATE	EXAMINE_EXPRESSION
005C	506	\$STRAN	!CK_EOS,EXAM_OK
005C	507	\$STRAN	!EXPRESSION
005C	508	\$STATE	
005C	509	\$STRAN	!CK_EOS,EXAM_OK
005C	510	\$STRAN	'/' EXAM_QUALS
005C	511	\$STRAN	SEMI,,,OPTSM_LENGTH,OPTIONS
005C	512	\$STRAN	COLON,,,OPTSM_RANGE,OPTIONS
005C	513	\$STATE	
005C	514	\$STRAN	!EXPRESSION
005C	515	\$STATE	
005C	516	\$STRAN	'/' EXAM_QUALS
005C	517	\$STRAN	TPAS_LAMBDA,EXAM_OK
005C	518	\$STATE	EXAM_QUAL_LOOP
005C	519	\$STRAN	!CK_EOS,EXAM_OK
005C	520	\$STRAN	'/'
005C	521	\$STATE	EXAM_QUALS
005C	522	\$STRAN	'INSTRUCTION',EXAM_QUAL_LOOP,,OPTSM_INST,OPTIONS
005C	523	\$STRAN	'NOSKIP',EXAM_QUAL_LOOP,,OPTSM_NOSKIP,OPTIONS
005C	524	\$STRAN	'PSL',EXAM_QUAL_LOOP,,OPTSM_PSL,OPTIONS
005C	525	\$STRAN	'TIME',EXAM_QUAL_LOOP,,OPTSM_TIME,OPTIONS
005C	526	\$STRAN	'CONDITION_VALUE',EXAM_QUAL_LOOP,,OPTSM_COND,OPTIONS
005C	527	\$STATE	EXAM_OK
005C	528	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,EXAM_MEMORY
005C	529	\$STATE	EXAM_INS
005C	530	\$STRAN	'/'
005C	531	\$STRAN	TPAS_LAMBDA,EXAMINE_EXPRESSION
005C	532	\$STATE	

```

005C 533          $STRAN 'NOSKIP',EXAM_INS,,OPTSM_NOSKIP,OPTIONS
005C 534          :
005C 535          : VALIDATE QUEUE expression
005C 536          :
005C 537          $STATE VALIDATE
005C 538          $STRAN 'QUEUE',VAL_QUEUE
005C 539
005C 540          $STATE VAL_QUEUE
005C 541          $STRAN !CK_EOS,VALIDATE_Q_OK
005C 542          $STRAN !VAL_QUEUE_OPTS,TPAS_EXIT
005C 543
005C 544          $STATE VAL_QUEUE_OPTS
005C 545          $STRAN '/',VAL_QUE_QUAL
005C 546          $STRAN !EXPRESSION,VAL_QUEUE
005C 547
005C 548          $STATE VAL_QUE_QUAL
005C 549          $STRAN 'SECF_RELATIVE',VAL_QUEUE,,OPTSM_SELF,OPTIONS
005C 550          $STRAN 'MAXIMUM_LINKS'
005C 551          $STATE
005C 552          $STRAN ':'
005C 553          $STRAN '='
005C 554          $STATE
005C 555          $STRAN !DECIMAL,VAL_QUEUE,VAL_SET_MAX
005C 556
005C 557          $STATE VALIDATE_Q_OK
005C 558          $STRAN TPAS_LAMBDA,TPAS_EXIT,VALIDATE_QUEUE
005C 559
005C 560          :
005C 561          : EXPRESSION ANALYSIS FOR EXAMINE ARGUMENTS
005C 562          :
005C 563          : THIS REDUCTION TABLE USES A SEPARATE STACK FOR PARSING
005C 564          : TOKENS AS THEY ENCOUNTERED. THE RESULT OF THE LAST
005C 565          : OPERATION IS ALWAYS PLACED IN TPASL_NUMBER TO RETURN
005C 566          : TO THE CALLING TRANSITION ROUTINE.
005C 567          :
005C 568
005C 569          $STATE EXPRESSION
005C 570          $STRAN !TERM
005C 571          $STATE EXPOP
005C 572          $STRAN '+',ADDITION
005C 573          $STRAN '-',SUBTRACTION
005C 574          $STRAN TPAS_LAMBDA,TPAS_EXIT
005C 575          $STATE ADDITION
005C 576          $STRAN !EXPRESSION,TPAS_EXIT,ADD2
005C 577          $STATE SUBTRACTION
005C 578          $STRAN !TERM, EXPOP, SUB2
005C 579          $STRAN !EXPRESSION,TPAS_EXIT,SUB2
005C 580          ;
005C 581          $STATE TERM
005C 582          $STRAN !FACTOR
005C 583          $STATE
005C 584          $STRAN '@',ARITH_SHIFT
005C 585          $STRAN '*',MULTIPLICATION
005C 586          $STRAN !DIVISION,TPAS_EXIT
005C 587          $STRAN TPAS_LAMBDA,TPAS_EXIT
005C 588          $STATE ARITH_SHIFT
005C 589          $STRAN !TERM,TPAS_EXIT,SHIFT2

```

005C	590	\$STATE	MULTIPLICATION
005C	591	\$STRAN	!TERM,TPAS_EXIT,MULT2
005C	592	\$STATE	DIVISION
005C	593	\$STRAN	'/'
005C	594	\$STATE	
005C	595	\$STRAN	!TERM,TPAS_EXIT,DIV2
005C	596		
005C	597	\$STATE	FACTOR
005C	598	\$STRAN	'-',NEGATE
005C	599	\$STRAN	'(',PARENS
005C	600	\$STRAN	'@',INDIRECT
005C	601	\$STRAN	!TPAS_EXIT,EVAL_DOT
005C	602	\$STRAN	TPAS_SYMBOL,TPAS_EXIT,EVAL_SYMBOL
005C	603	\$STRAN	!CONSTANT,TPAS_EXIT,EVAL_CONSTANT
005C	604	\$STATE	NEGATE
005C	605	\$STRAN	!FACTOR,TPAS_EXIT,NEG1
005C	606	\$STATE	INDIRECT
005C	607	\$STRAN	!FACTOR,TPAS_EXIT,EVAL_CONTENTS
005C	608	\$STATE	PARENS
005C	609	\$STRAN	!EXPRESSION
005C	610	\$STATE	
005C	611	\$STRAN	')',TPAS_EXIT
005C	612		
005C	613	\$STATE	CONSTANT
005C	614	\$STRAN	'+',CONSTANT
005C	615	\$STRAN	'^',RADIX
005C	616	\$STRAN	!HEX,TPAS_EXIT
005C	617		
005C	618	\$STATE	RADIX
005C	619	\$STRAN	'O',OCTAL
005C	620	\$STRAN	'X',HEX
005C	621	\$STRAN	'D',DECIMAL
005C	622		
005C	623	\$STATE	OCTAL
005C	624	\$STRAN	TPAS_OCTAL,TPAS_EXIT
005C	625		
005C	626	\$STATE	DECIMAL
005C	627	\$STRAN	TPAS_DECIMAL,TPAS_EXIT
005C	628		
005C	629	\$STATE	HEX
005C	630	\$STRAN	'G',SYSREG
005C	631	\$STRAN	'H',CTLREG
005C	632	\$STRAN	TPAS_HEX,TPAS_EXIT
005C	633	\$STATE	SYSREG
005C	634	\$STRAN	TPAS_HEX,TPAS_EXIT,ADD8000
005C	635	\$STATE	CTLREG
005C	636	\$STRAN	TPAS_HEX,TPAS_EXIT,ADD7FFE
005C	637		
005C	638	:	
005C	639	:	SHOW PROCESS [name/ALL] [/INDEX=n] [/SYSTEM]
005C	640	:	/WORKING_SET_LIST [or WSL]
005C	641	:	/PROCESS_SECTION_TABLE [or PST]
005C	642	:	/PAGE_TABLES [or PPT]
005C	643	:	start:end
005C	644	:	start:length
005C	645	:	/P0
005C	646	:	/P1

```

005C 647 : /REGISTERS
005C 648 : /PCB
005C 649 : /PHD
005C 650 : /ALL
005C 651 : /RMS [=opt]
005C 652 : [= (opt1,opt2,....,optn)]
005C 653 : /LOCKS
005C 654 : /CHANNELS
005C 655 :
005C 656 :
005C 657 $STATE PROCESS
005C 658 $STRAN !CK_EOS,TPAS_EXIT,SHOW_PROCESS
005C 659 $STRAN !/
005C 660 $STRAN TPAS_SYMBOL,PROCESS,,,PROC_NAME
005C 661 $STATE PROCESS1
005C 662 $STRAN 'INDEX',PROC_PIX
005C 663 $STRAN 'SYSTEM',PROCESS,,OPTSM_SYSPROC,OPTIONS
005C 664 $STRAN 'WORKING SET LIST',PROCESS,,OPTSM_WSL,OPTIONS
005C 665 $STRAN 'WSL',PROCESS,,OPTSM_WSL,OPTIONS
005C 666 $STRAN 'PROCESS SECTION TABLE',PROCESS,,OPTSM_PST,OPTIONS
005C 667 $STRAN 'PST',PROCESS,,OPTSM_PST,OPTIONS
005C 668 $STRAN 'PAGE TABLES',PROC_PPT,,OPTSM_PPT,OPTIONS
005C 669 $STRAN 'PPT',PROC_PPT,,OPTSM_PPT,OPTIONS
005C 670 $STRAN 'P0',PROCESS,,<OPTSM_PPT!OPTSM_P0_PPT>,OPTIONS
005C 671 $STRAN 'P1',PROCESS,,<OPTSM_PPT!OPTSM_P1_PPT>,OPTIONS
005C 672 $STRAN 'REGISTERS',PROCESS,,OPTSM_REGS,OPTIONS
005C 673 $STRAN 'PCB',PROCESS,,OPTSM_PCB,OPTIONS
005C 674 $STRAN 'PHD',PROCESS,,OPTSM_PHD,OPTIONS
005C 675 $STRAN 'ALL',PROCESS,,<OPTSM_PCB!OPTSM_PHD!OPTSM_REGS!-
005C 676 OPTSM_WSL!OPTSM_PST!OPTSM_PPT!OPTSM_CHAN>,OPTIONS
005C 677 $STRAN 'LOCKS',PROCESS,,OPTSM_LCK,OPTIONS
005C 678 $STRAN 'CHANNELS',PROCESS,,OPTSM_CHAN,OPTIONS
005C 679 $STRAN !RMS,PROCESS,,OPTSM_RMS,OPTIONS
005C 680 $STATE PROC_PIX
005C 681 $STRAN ':'
005C 682 $STRAN '='
005C 683 $STATE
005C 684 $STRAN TPAS_HEX,PROCESS,STORE_PROC_INDEX
005C 685 $STATE PROC_PPT
005C 686 $STRAN !CK_EOS,TPAS_EXIT,SHOW_PROCESS
005C 687 $STRAN !/,PROCESS1
005C 688 $STRAN !EXPRESSION
005C 689 $STATE
005C 690 $STRAN SEMI,,,OPTSM_PPT_LEN,OPTIONS
005C 691 $STRAN COLON,,,OPTSM_PPT_RNG,OPTIONS
005C 692 $STATE
005C 693 $STRAN !EXPRESSION,PROCESS
005C 694 :
005C 695 :
005C 696 : SHOW PROCESS/RMS [=opt]
005C 697 : [= (opt1,opt2,....,optn)]
005C 698 :
005C 699 :
005C 700 $STATE RMS
005C 701 $STRAN 'RMS'
005C 702 $STATE
005C 703 $STRAN !DISPLAY,TPAS_EXIT,,OPTSM_RMSD,OPTIONS

```

005C	704	\$STRAN	TPAS_LAMBDA,TPAS_EXIT
005C	705		
005C	706	:	
005C	707	:	
005C	708	:	SET PROCESS name /INDEX=n /SYSTEM
005C	709	:	
005C	710	\$STATE	SET PROC
005C	711	\$STRAN	!INDEX
005C	712	\$STRAN	TPAS_SYMBOL,,,,PROC_NAME
005C	713	\$STATE	
005C	714	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,SET_PROCESS
005C	715		
005C	716	\$STATE	INDEX
005C	717	\$STRAN	!/'
005C	718	\$STATE	
005C	719	\$STRAN	'INDEX'
005C	720	\$STRAN	'SYSTEM',TPAS_EXIT,,OPTSM_SYSPROC,OPTIONS
005C	721	\$STATE	
005C	722	\$STRAN	!:'
005C	723	\$STRAN	!:'
005C	724	\$STATE	
005C	725	\$STRAN	TPAS_HEX,TPAS_EXIT,STORE_PROC_INDEX
005C	726		
005C	727	:	
005C	728	:	SHOW POOL /IRP /NONPAGED /PAGED /ALL /SUMMARY /HEADER
005C	729	:	
005C	730	:	
005C	731	\$STATE	POOL
005C	732	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_POOL
005C	733	\$STRAN	!/'
005C	734	\$STRAN	TPAS_LAMBDA,POOL_RANGE
005C	735	\$STATE	
005C	736	\$STRAN	'IRP',POOL,,OPTSM_IRP,OPTIONS
005C	737	\$STRAN	'LRP',POOL,,OPTSM_LRP,OPTIONS
005C	738	\$STRAN	'SRP',POOL,,OPTSM_SRP,OPTIONS
005C	739	\$STRAN	'NONPAGED',POOL,,OPTSM_NONPAGED,OPTIONS
005C	740	\$STRAN	'PAGED',POOL,,OPTSM_PAGED,OPTIONS
005C	741	\$STRAN	'ALL',POOL,,OPTSM_IRP!OPTSM_NONPAGED!OPTSM_PAGED,OPTIONS
005C	742	\$STRAN	'SUMMARY',POOL,,OPTSM_SUMMARY,OPTIONS
005C	743	\$STRAN	'HEADER',POOL,,OPTSM_HEADER,OPTIONS
005C	744	\$STRAN	'FREE',POOL,,OPTSM_FREE,OPTIONS
005C	745	\$STRAN	'TYPE',,,,OPTSM_TYPE,OPTIONS
005C	746	\$STATE	
005C	747	\$STRAN	!:'
005C	748	\$STRAN	!:'
005C	749	\$STATE	
005C	750	\$STRAN	TPAS_SYMBOL,POOL,,,STRUCTURE
005C	751		
005C	752	:	
005C	753	:	SHOW POOL start:end
005C	754	:	start:length
005C	755	:	
005C	756	:	
005C	757	\$STATE	POOL_RANGE
005C	758	\$STRAN	!EXPRESSION
005C	759	\$STATE	
005C	760	\$STRAN	SEMI,,,OPTSM_LENGTH,OPTIONS

005C	761	\$STRAN	COLON,,,OPT\$M_RANGE,OPTIONS
005C	762	\$STATE	
005C	763	\$STRAN	!EXPRESSION
005C	764	\$STATE	
005C	765	\$STRAN	!CK_EOS,TPAS_EXIT,SHOW_POOL_RANGE
005C	766		
005C	767	:	
005C	768	:	
005C	769	:	
005C	770		SHOW STACK /mode
005C	771	\$STATE	STACK
005C	772	\$STRAN	!CK_EOS,TPAS_EXIT,PRINT_STACKS
005C	773	\$STRAN	!/'
005C	774	\$STRAN	TPAS_LAMBDA,STACK_RANGE
005C	775	\$STATE	
005C	776	\$STRAN	'INTERRUPT',STACK,,OPT\$M_ISP,OPTIONS
005C	777	\$STRAN	'KERNEL',STACK,,OPT\$M_KSP,OPTIONS
005C	778	\$STRAN	'EXECUTIVE',STACK,,OPT\$M_ESP,OPTIONS
005C	779	\$STRAN	'SUPERVISOR',STACK,,OPT\$M_SSP,OPTIONS
005C	780	\$STRAN	'USER',STACK,,OPT\$M_USP,OPTIONS
005C	781	\$STRAN	'ALL',STACK,,OPT\$M_ALL,OPTIONS
005C	782		
005C	783	:	
005C	784	:	
005C	785	:	
005C	786	:	
005C	787		SHOW STACK start:end
005C	788		start:length
005C	789	\$STATE	STACK_RANGE
005C	790	\$STRAN	!EXPRESSION
005C	791	\$STATE	
005C	792	\$STRAN	SEMI,,,OPT\$M_LENGTH,OPTIONS
005C	793	\$STRAN	COLON,,,OPT\$M_RANGE,OPTIONS
005C	794	\$STATE	
005C	795	\$STRAN	!EXPRESSION
005C	796	\$STATE	
005C	797	\$STRAN	!CK_EOS,TPAS_EXIT,PRINT_ANY_STACK
005C	798		
005C	799	:	
005C	800	:	
005C	801		SHOW PFN_DATA <optional pfn>
005C	802	\$STATE	PFN_DATA
005C	803	\$STRAN	!/'-PFN_OPTION
005C	804	\$STRAN	!EXPRESSION,,,OPT\$M_SINGLEPFN,OPTIONS
005C	805	\$STRAN	TPAS_LAMBDA,,,OPT\$M_FREE!OPT\$M_MODIFIED!OPT\$M_BAD!OPT\$M_WHOLEPFN,OPT
005C	806	\$STATE	
005C	807	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,DISPLAY_PFN
005C	808		
005C	809	\$STATE	PFN_OPTS
005C	810	\$STRAN	!/'
005C	811	\$STRAN	TPAS_LAMBDA,TPAS_EXIT,DISPLAY_PFN
005C	812	\$STATE	PFN_OPTION
005C	813	\$STRAN	'FREE',PFN_OPTS,,OPT\$M_FREE,OPTIONS
005C	814	\$STRAN	'MODIFIED',PFN_OPTS,,OPT\$M_MODIFIED,OPTIONS
005C	815	\$STRAN	'BAD',PFN_OPTS,,OPT\$M_BAD,OPTIONS
005C	816	\$STRAN	'SYSTEM',PFN_OPTS,,OPT\$M_WHOLEPFN,OPTIONS
005C	817	\$STRAN	'ALL',PFN_OPTS,,OPT\$M_FREE!OPT\$M_MODIFIED!OPT\$M_BAD!OPT\$M_WHOLEPFN,O

```

005C 818
005C 819 :
005C 820 :
005C 821 :
005C 822 :
005C 823 $STATE PAGE_TABLE
005C 824 $STRAN '/' PAGE_OPTION
005C 825 $STRAN TPAS_LAMBDA,,,OPTSM_SYSTEM!OPTSM_GLOBAL,OPTIONS
005C 826 $STATE
005C 827 $STRAN !CK_EOS,TPAS_EXIT,DISPLAY_SPT
005C 828 $STRAN TPAS_LAMBDA,PAGE_RANGE
005C 829
005C 830 $STATE PAGE_OPTS
005C 831 $STRAN '/'
005C 832 $STRAN !CK_EOS,TPAS_EXIT,DISPLAY_SPT
005C 833 $STRAN TPAS_LAMBDA,PAGE_RANGE
005C 834 $STATE PAGE_OPTION
005C 835 $STRAN 'SYSTEM',PAGE_OPTS,,,OPTSM_SYSTEM,OPTIONS
005C 836 $STRAN 'GLOBAL',PAGE_OPTS,,,OPTSM_GLOBAL,OPTIONS
005C 837 $STRAN 'ALL',PAGE_OPTS,,,OPTSM_ALL,OPTIONS
005C 838
005C 839
005C 840 :
005C 841 :
005C 842 :
005C 843 :
005C 844 :
005C 845 $STATE PAGE_RANGE
005C 846 $STRAN !EXPRESSION
005C 847 $STATE
005C 848 $STRAN SEMI,,,OPTSM_LENGTH,OPTIONS
005C 849 $STRAN COLON,,,OPTSM_RANGE,OPTIONS
005C 850 $STATE
005C 851 $STRAN !EXPRESSION
005C 852 $STATE
005C 853 $STRAN !CK_EOS,TPAS_EXIT,DISPLAY_SPT_RANGE
005C 854
005C 855 :
005C 856 :
005C 857 :
005C 858 :
005C 859 $STATE SHOW_SUMMARY
005C 860 $STRAN !CK_EOS,TPAS_EXIT,PROCESS_SUMMARY
005C 861 $STRAN '/'
005C 862 $STATE
005C 863 $STRAN 'IMAGE',SHOW_SUMMARY,,,OPTSM_IMAGE,OPTIONS
005C 864
005C 865 :
005C 866 :
005C 867 :
005C 868 :
005C 869 $STATE SYMBOL
005C 870 $STRAN TPAS_LAMBDA,,CLEAR_SYMBOL_NAME
005C 871 $STATE SYMBOL_QUALS
005C 872 $STRAN '/' SYMBOL_OPTS
005C 873 $STRAN TPAS_SYMBOL_SYM,,,SYMBOL_NAME
005C 874 $STRAN !CK_EOS,TPAS_EXIT,SHOW_SYMBOL

```

```

005C 875 $STATE SYM
005C 876 $STRAN '/'
005C 877 $STRAN !CK_EOS,TPAS_EXIT,SHOW_SYMBOL
005C 878 $STATE SYMBOL_OPTS
005C 879 $STRAN 'ALL',SYMBOL_QUALS,,OPTSM_ALL,OPTIONS
005C 880
005C 881 :
005C 882 : SHOW DEVICE device
005C 883 :
005C 884
005C 885 $STATE DEVICE
005C 886 $STRAN '/' DEVICE_OPTS
005C 887 $STRAN TPAS_SYMBOL,TPAS_EXIT,DISPLAY_DEVICE
005C 888 $STRAN TPAS_LAMBDA,TPAS_EXIT,DISPLAY_DEVICE
005C 889
005C 890 $STATE DEVICE_OPTS
005C 891 $STRAN 'ADDRESS'
005C 892 $STATE
005C 893 $STRAN '='
005C 894 $STATE
005C 895 $STRAN !EXPRESSION
005C 896 $STATE
005C 897 $STRAN TPAS_LAMBDA,TPAS_EXIT,DISPLAY_DEVBYADDR
005C 898
005C 899 :
005C 900 : DEFINE symbol = value
005C 901 :
005C 902
005C 903 $STATE DEFINE
005C 904 $STRAN '/' TPAS_EXIT,DEFINE_KEY
005C 905 $STRAN TPAS_SYMBOL,,,,SYMBOL_DESC
005C 906 $STATE
005C 907 $STRAN '='
005C 908 $STRAN TPAS_LAMBDA
005C 909 $STATE
005C 910 $STRAN !EXPRESSION
005C 911 $STATE
005C 912 $STRAN TPAS_LAMBDA,TPAS_EXIT,DEFINE_SYMBOL
005C 913
005C 914 :
005C 915 : HELP COMMAND
005C 916 :
005C 917
005C 918 $STATE HELP_CMD
005C 919 $STRAN !CK_EOS,TPAS_EXIT,DISPLAY_HELP
005C 920 $STRAN !COLLECT,TPAS_EXIT,DISPLAY_HELP
005C 921
005C 922 :
005C 923 : EXIT COMMAND
005C 924 :
005C 925
005C 926 $STATE EXIT_CMD
005C 927 $STRAN !CK_EOS,TPAS_EXIT,EXIT_COMMAND
005C 928 $STRAN TPAS_STRING,TPAS_EXIT,EXIT_COMMAND
005C 929
005C 930 :
005C 931 : READ filespec

```

```

005C 932 ;
005C 933 ;
005C 934 $STATE READ_CMD
005C 935 $STRAN !READ_QUALS
005C 936 $STRAN TPAS_LAMBDA
005C 937 $STATE
005C 938 $STRAN !COLLECT,,,,FILE_DESC
005C 939 $STATE
005C 940 $STRAN TPAS_LAMBDA,TPAS_EXIT,READ_SYMFIL
005C 941
005C 942 $STATE READ_QUALS
005C 943 $STRAN '/'
005C 944 $STATE
005C 945 $STRAN 'RELOCATE'
005C 946 $STATE
005C 947 $STRAN '='
005C 948 $STATE
005C 949 $STRAN !EXPRESSION,TPAS_EXIT,STORE_RELOCATION
005C 950
005C 951 ;
005C 952 ;
005C 953 ;
005C 954 COPY filespec
005C 955 $STATE COPY_CMD
005C 956 $STRAN !COLLECT,,,,FILE_DESC
005C 957 $STATE
005C 958 $STRAN TPAS_LAMBDA,TPAS_EXIT,SAVE_DUMP
005C 959
005C 960 ;
005C 961 ;
005C 962 ;
005C 963
005C 964 $STATE FORMAT_CMD
005C 965 $STRAN !FORMAT_OPTS
005C 966 $STATE
005C 967 $STRAN !EXPRESSION
005C 968 $STATE
005C 969 $STRAN !FORMAT_OPTS
005C 970 $STATE
005C 971 $STRAN TPAS_LAMBDA,TPAS_EXIT,FORMAT
005C 972
005C 973 $STATE FORMAT_OPTS
005C 974 $STRAN '/'
005C 975 $STRAN TPAS_LAMBDA,TPAS_EXIT
005C 976 $STATE
005C 977 $STRAN 'TYPE'
005C 978 $STATE
005C 979 $STRAN '='
005C 980 $STRAN ':'
005C 981 $STATE
005C 982 $STRAN TPAS_SYMBOL,,,,STRUCTURE
005C 983 $STATE
005C 984 $STRAN TPAS_LAMBDA,TPAS_EXIT,,1,OPTIONS
005C 985
005C 986
005C 987 $END_STATE
005C 988

```

PARSE
V04-000

Parse tables for SDA commands
PARSER STATE TABLE

J 1

16-SEP-1984 01:36:01
5-SEP-1984 03:33:21

VAX/VMS Macro V04-00
[SDA.SRC]PARSE.MAR;1

Page 21
(4)

00000000 989 .PSECT PARSE,EXE,NOWRT

PAR
Sym
SET
SET
SHI
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SMG
SMG
SMG
SNA
STA
STA
STA
STO
STO
STO
STO
STO
STO
STO
STO
STR
SUB
SUB
SYM
SYM
SYM
SYM
SYM
SYM
SYM
SYS
TER
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA

```

0000 991 .SBTTL EXPRESSION ANALYSIS ACTION ROUTINES
0000 992 :---
0000 993 :
0000 994 EXPRESSION ANALYSIS ACTION ROUTINES
0000 995 :
0000 996 THESE ROUTINES ARE CALLED FROM THE EXPRESSION ANALYSIS
0000 997 PARSING TRANSITIONS TO EVALUATE ARGUMENTS USING A SEPARATE
0000 998 STACK SPACE.
0000 999 :
0000 1000 INPUTS:
0000 1001 :
0000 1002 AP = ADDRESS OF TPARSE PARAMETER BLOCK
0000 1003 :
0000 1004 :---
0000 1005 :
61 44 0000 1006 ADD2: .WORD 0
81 10 0002 1007 BSBB EVAL_SETUP
47 C0 0004 1008 ADDL2 (R1)+,(R1) ; ADD 2 ARGUMENTS, RESULT ON TOP
11 0007 1009 BRB STORE
0009 1010 :
61 3B 0000 1011 MULT2: .WORD 0
81 10 000B 1012 BSBB EVAL_SETUP
3E C4 000D 1013 MULL2 (R1)+,(R1) ; MULTIPLY 2 ARGUMENTS, RESULT ON TOP
11 0010 1014 BRB STORE
0012 1015 :
61 32 0000 1016 SUB2: .WORD 0
81 10 0014 1017 BSBB EVAL_SETUP
35 C2 0016 1018 SUBL2 (R1)+,(R1) ; SUBTRACT 2 ARGUMENTS, RESULT ON TOP
11 0019 1019 BRB STORE
001B 1020 :
61 29 0000 1021 DIV2: .WORD 0
81 10 001D 1022 BSBB EVAL_SETUP
3C D5 001F 1023 TSTL (R1) ; DO NOT ALLOW DIVIDE BY ZERO
11 0021 1024 BEQL FAILURE
61 81 C6 0023 1025 DIVL2 (R1)+,(R1) ; DIVIDE 2 ARGUMENTS, RESULT ON TOP
28 11 0026 1026 BRB STORE
0028 1027 :
61 50 0000 1028 SHIFT2: .WORD 0
61 1C 10 002A 1029 BSBB EVAL_SETUP
50 D0 002C 1030 MOVL (R1)+,R0 ; GET SHIFT COUNT
1B 78 002F 1031 ASHL R0,(R1),(R1) ; SHIFT LEFT BY COUNT
11 0033 1032 BRB STORE
0035 1033 :
61 0F 0000 1034 NEG1: .WORD 0
61 10 0037 1035 BSBB EVAL_SETUP
CE 0039 1036 MNEGL (R1)+,(R1) ; NEGATE 1 ARGUMENT, RESULT ON TOP
12 11 003C 1037 BRB STORE
003E 1038 :
003E 1039 EVAL_CONSTANT:
0000 003E 1040 .WORD 0
71 0C 10 0040 1041 BSBB EVAL_SETUP
1C AC D0 0042 1042 MOVL TPA$C_NUMBER(AP),-(R1) ; PUSH RESULT FROM TPARSE
08 11 0046 1043 BRB STORE
0048 1044 :
51 00000000*EF D0 0048 1045 EVAL_SETUP:
05 004F 1046 MOVL ESP,R1 ; R1 = STACK POINTER
004F 1047 RSB

```

			0050	1048		
			0050	1049	STORE:	
1C AC	61	D0	0050	1050	MOVL	(R1),TPA\$L_NUMBER(AP) ; SAVE RESULT FOR CALLER
00000000'EF	51	D0	0054	1051	MOVL	R1,ESP ; STORE STACK POINTER
	01	D0	005B	1052	MOVL	#1,R0
		04	005E	1053	RET	
			005F	1054		
			005F	1055	FAILURE:	
	50	D4	005F	1056	CLRL	R0
		04	0061	1057	RET	
			0062	1058		
			0062	1059	EVAL_SYMBOL:	
		0000	0062	1060	.WORD	0
	10 AC	7F	0064	1061	PUSHAQ	TPA\$L_TOKENCNT(AP) ; DESCRIPTOR OF SYMBOL NAME
00000000'EF	01	FB	0067	1062	CALLS	#1,SYMBOL_VALUE ; TRY TO FIND IN SYMBOL TABLE
EE	50	E9	006E	1063	BLBC	R0,FAILURE
	51	DD	0071	1064	PUSHL	R1
	D3	10	0073	1065	BSBB	EVAL_SETUP
	71	8ED0	0075	1066	POPL	-(R1)
	D6	11	0078	1067	BRB	STORE ; PUSH RESULT ONTO STACK
			007A	1068		
			007A	1069	EVAL_DOT:	
		0000	007A	1070	.WORD	0
	CA	10	007C	1071	BSBB	EVAL_SETUP
71	00000000'EF	D0	007E	1072	MOVL	ADDRESS,-(R1)
	C9	11	0085	1073	BRB	STORE
			0087	1074		
			0087	1075	EVAL_CONTENTS:	
		0000	0087	1076	.WORD	0
	BD	10	0089	1077	BSBB	EVAL_SETUP
			008B	1078	REQMEM	@(R1)
	51	DD	0095	1079	PUSHL	R1 ; GET CONTENTS OF LOCATION
	AF	10	0097	1080	BSBB	EVAL_SETUP
	61	8ED0	0099	1081	POPL	(R1) ; AND REPLACE ARG. ON TOP OF STACK
	B2	11	009C	1082	BRB	STORE
			009E	1083		
			009E	1084	ADD8000:	; ADDRESS IN SYSTEM REGION
		0000	009E	1085	.WORD	0
1C AC	80000000	8F	00A0	1086	ADDL	#^X80000000,TPA\$L_NUMBER(AP) ; ADD BASE OF SYSTEM REGION
		04	00A8	1087	RET	
			00A9	1088		
			00A9	1089	ADD7FFE:	; ADDRESS IN CONTROL REGION
		0000	00A9	1090	.WORD	0
1C AC	7FFE0000	8F	00AB	1091	ADDL	#^X7FFE0000,TPA\$L_NUMBER(AP) ; CONTROL REGION
		04	00B3	1092	RET	

		00B4	1094	.SBTTL	STORE_RELOCATION, STORE_READ/RELOCATE VALUE
		00B4	1095	---	
		00B4	1096	:	
		00B4	1097	:	STORE THE VALUE OF THE READ/RELOCATE QUALIFIER.
		00B4	1098	:	
		00B4	1099	:	INPUTS:
		00B4	1100	:	
		00B4	1101	:	TPA\$L_NUMBER(AP) = RELOCATION BASE
		00B4	1102	:	
		00B4	1103	:	OUTPUTS:
		00B4	1104	:	
		00B4	1105	:	RELOCATE_BASE = RELOCATION BASE
		00B4	1106	---	
		00B4	1107	:	
		00B4	1108	:	STORE_RELOCATION:
	0000	00B4	1109	:	.WORD 0
		00B6	1110	:	
00000000'EF	1C AC	D0	00B6	1111	MOVL TPA\$L_NUMBER(AP),RELOCATE_BASE
		04	00BE	1112	RET
			00BF	1113	
			00BF	1114	.SBTTL STORE_PROC_INDEX, STORE PROCESS INDEX LOW BITS
			00BF	1115	---
			00BF	1116	:
			00BF	1117	:
			00BF	1118	:
			00BF	1119	:
			00BF	1120	:
			00BF	1121	:
			00BF	1122	:
			00BF	1123	:
			00BF	1124	:
			00BF	1125	:
			00BF	1126	:
			00BF	1127	---
			00BF	1128	:
			00BF	1129	:
	0000		00BF	1130	:.WORD 0
			00C1	1131	:
1C AC	00000000'EF	00	00C1	1132	EXTZV #0, PIX_WIDTH, TPA\$L_NUMBER(AP), PROC_INDEX
	00000000'EF	EF	00CA		
	50	01	00CF	1133	MOVL #1, R0
		D0	00D2	1134	
		04	00D2	1135	RET

```

00D3 1137 .SBTTL SAVE_COMMAND, SAVE THE CURRENT COMMAND
00D3 1138 :---
00D3 1139 :
00D3 1140 : SAVE THE CURRENT COMMAND IN THE COMMAND BUFFER
00D3 1141 : SO THAT A LATER 'REPEAT' COMMAND CAN RECALL IT.
00D3 1142 :
00D3 1143 : INPUTS:
00D3 1144 :
00D3 1145 : TPA$L_STRINGCNT(AP) = DESCRIPTOR OF STRING
00D3 1146 :
00D3 1147 : OUTPUTS:
00D3 1148 :
00D3 1149 : PREV_COMMAND = DESCRIPTOR OF SAVED STRING
00D3 1150 :
00D3 1151 :---
00D3 1152 :
00D3 1153 SAVE_COMMAND::
003C 00D3 1154 .WORD ^M<R2,R3,R4,R5>
00D5 1155
00D5 1156 MOVW TPA$L_STRINGCNT(AP),PREV_COMMAND
00DD 1157 MOVC TPA$L_STRINGCNT(AP),@TPA$L_STRINGPTR(AP),-
00E2 1158 @PREV_COMMAND+4
00E7 1159
00E7 1160 ; Tell the SMG$ level how to repeat this command
00E7 1161
00E7 1162 CLRL -(SP) ; State string null
00E9 1163 PUSHAB PREV_COMMAND ; Last command
00EF 1164 PUSHAB KEY_ATTR ; attributes
00F5 1165 CLRL -(SP) ; No IF STATE
00F7 1166 PUSHAB REPEAT_KEY ; Repeat key name (KPO defaults)
00FD 1167 PUSHAB KEYTABLE ; key table
0103 1168 CALLS #6,G^SMG$ADD_KEY_DEF ; Change key definition
010A 1169 SIGNAL
0116 1170 RET

```

```

0117 1172 .SBTTL REPEAT_COMMAND, REPEAT THE PREVIOUS COMMAND
0117 1173 :---
0117 1174 :
0117 1175 : STORE THE PREVIOUS COMMAND IN THE COMMAND BUFFER
0117 1176 : AND RESTART THE PARSE.
0117 1177 :
0117 1178 : INPUTS:
0117 1179 :
0117 1180 : PREV_COMMAND = DESCRIPTOR OF PREVIOUS COMMAND
0117 1181 :
0117 1182 : OUTPUTS:
0117 1183 :
0117 1184 : TPASL_STRINGCNT(AP) = THE PREVIOUS COMMAND IS RELOADED
0117 1185 :
0117 1186 :---
0117 1187
0117 1188 REPEAT_COMMAND::
0117 1189 .WORD 0
0119 1190
0119 1191 MOVQ PREV_COMMAND,TPASL_STRINGCNT(AP)
0121 1192 RET

```

0000 00000050'EF 7D 04

```

0122 1194 .SBTTL DEFINE_KEY - DEFINE KEY
0122 1195 :+++
0122 1196 :
0122 1197 :
0122 1198 :
0122 1199 :---
0122 1200
0004 0122 1201 .ENTRY DEFINE_KEY,*M<R2>
0124 1202
0124 1203 PUSHAL INPUT_BUFFER ; address
7E 00000000'EF DF 012A 1204 MOVZWL INPUT_LEN,-(SP) ; length
00000000'EF 3C 0131 1205 PUSH SP ; address of desc
5E DD 0133 1206 PUSHAB KEYTABLE ; table id
00000000'EF 9F 0139 1207 CALLS #2,G^SMG$DEFINE_KEY ; Define key...
00000000'GF 02 FB 0140 1208 ADDL #8,SP ; pop
5E 08 C0 0143 1209 SIGNAL
014F 1210 STATUS SUCCESS ; ok
04 0156 1211 RET
0157 1212

```

```

0157 1214 .SBTTL Action Routines for SET:SHOW RMS/DISPLAY=optionspec
0157 1215 :+++
0157 1216 : SETUP_RMS_TMP -- Setup the temporary RMS options word.
0157 1217 :
0157 1218 : Inputs:
0157 1219 :
0157 1220 : None.
0157 1221 :
0157 1222 : Outputs:
0157 1223 :
0157 1224 : RMS_DIS_TMP = 0
0157 1225 : RMS_IFI_TMP = 0
0157 1226 : ---
0157 1227 :
00000000'EF 0000 0157 1228 SETUP_RMS_TMP: .WORD 0
00000000'EF D4 0159 1229 CRL RMS_DIS_TMP
50 01 B4 015F 1230 CLRW RMS_IFI_TMP
04 D0 0165 1231 MOVL #1,R0
0168 1232 RET
0169 1233
0169 1234 :+++
0169 1235 : STORE_TMP1_OPT -- Merge this particular option into temporary option word.
0169 1236 :
0169 1237 : Inputs:
0169 1238 :
0169 1239 : RMS_DIS_TMP1 = Outcome of !DIS_OPT state (Possibly containing OPT$M_NO)
0169 1240 :
0169 1241 : Outputs:
0169 1242 :
0169 1243 : RMS_DIS_TMP is set or cleared with value of RMS_DIS_TMP1 depending
0169 1244 : on value of OPT$M_NO in RMS_DIS_TMPT.
0169 1245 : ---
0169 1246 :
00000000'EF 0000 0169 1247 STORE_TMP1_OPT: .WORD 0
00000000'EF 01 D3 016B 1248 BITL #OPT$M_NO,RMS_DIS_TMP1 ; was this display option 'no'ed'?
00000000'EF 00000000'EF 0F 12 0172 1249 BNEQ 10$ ; if neq yes
50 01 C8 0174 1250 BISL RMS_DIS_TMP1,RMS_DIS_TMP
00000000'EF 00000000'EF D0 017F 1251 MOVL #1,R0
50 01 04 0182 1252 RET
00000000'EF 00000000'EF CA 0183 1253 10$: BICL RMS_DIS_TMP1,RMS_DIS_TMP
50 01 D0 018E 1254 MOVL #1,R0
04 0191 1255 RET
0192 1256
0192 1257 :+++
0192 1258 : END_SET_RMS -- Finish the SET RMS Command.
0192 1259 :
0192 1260 : Inputs:
0192 1261 :
0192 1262 : Final outputs of the SET RMS command:
0192 1263 : RMS_DIS_TMP
0192 1264 : RMS_IFI_TMP
0192 1265 :
0192 1266 : Outputs:
0192 1267 :
0192 1268 : Permanent display options set:
0192 1269 : RMS_DIS_OPT = RMS_DIS_TMP
0192 1270 : RMS_IFI = RMS_IFI_TMP

```

```

00000000'EF 00000000'EF 0000 0192 1271 ;---
00000000'EF 00000000'EF 0000 0192 1272
50 01 D0 0192 1273 END_SET_RMS:.WORD 0
B0 0194 1274 MOVW RMS_DIS_TMP,RMS_DIS_OPT
D0 019F 1275 MOVW RMS_IFI_TMP,RMS_IFI-
04 01AA 1276 MOVL #1,R0
01AD 1277 RET
01AE 1278
01AE 1279 ;+++
01AE 1280 : STORE_IFI_TMP -- Set the temporary IFI from the results of !EXPRESSION.
01AE 1281 :
01AE 1282 : Inputs:
01AE 1283 :
01AE 1284 : TPA$$_NUMBER(AP) = Results of !EXPRESSION
01AE 1285 :
01AE 1286 : Outputs:
01AE 1287 :
01AE 1288 : RMS_IFI_TMP contains above value.
01AE 1289 :---
01AE 1290
00000000'EF 1C AC 0000 01AE 1291 STORE_IFI_TMP:.WORD 0
50 01 B0 01B0 1292 MOVW TPA$$_NUMBER(AP),RMS_IFI_TMP
D0 01B8 1293 MOVL #1,R0
04 01BB 1294 RET
01BC 1295
01BC 1296 ;+++
01BC 1297 : CLR_IFI_TMP -- Clear RMS_IFI_TMP.
01BC 1298 :
01BC 1299 : Inputs:
01BC 1300 :
01BC 1301 : None.
01BC 1302 :
01BC 1303 : Outputs:
01BC 1304 :
01BC 1305 : RMS_IFI_TMP = 0
01BC 1306 :---
01BC 1307
00000000'EF 0000 01BC 1308 CLR_IFI_TMP:.WORD 0
50 01 B4 01BE 1309 CLRW RMS_IFI_TMP
D0 01C4 1310 MOVL #1,R0
04 01C7 1311 RET
01C8 1312
01C8 1313 ;+++
01C8 1314 : SETUP_DIS_OPT -- Setup for !DIS_OPT state.
01C8 1315 :
01C8 1316 : Inputs:
01C8 1317 :
01C8 1318 : None.
01C8 1319 :
01C8 1320 : Outputs:
01C8 1321 :
01C8 1322 : RMS_DIS_TMP1 = 0
01C8 1323 :---
01C8 1324
00000000'EF 0000 01C8 1325 SETUP_DIS_OPT:.WORD 0
50 01 D4 01CA 1326 CLRW RMS_DIS_TMP1
D0 01D0 1327 MOVL #1,R0

```

```

04 01D3 1328      RET
    01D4 1329
    01D4 1330 :+++
    01D4 1331 : STORE_STAR -- Include into temporary options previous perm options.
    01D4 1332 :
    01D4 1333 : Inputs:
    01D4 1334 :
    01D4 1335 :     RMS_DIS_OPT = Permanent options.
    01D4 1336 :     RMS_IFI = Permanent IFI.
    01D4 1337 :
    01D4 1338 : Outputs:
    01D4 1339 :
    01D4 1340 :     RMS_DIS_TMP1 = RMS_DIS_OPT - OPT$M_NO
    01D4 1341 :     RMS_IFI_TMP = RMS_IFI
    01D4 1342 : ---
    01D4 1343
    0000 01D4 1344 STORE_STAR:.WORD 0
    00000000'EF DD 01D6 1345      PUSHL  RMS_DIS_OPT
    6E 01 CA 01DC 1346      BICL    #OPT$M_NO,(SP)
    00000000'EF 8ED0 01DF 1347      POPL   RMS_DIS_TMP1
    00000000'EF B0 01E6 1348      MOVW   RMS_IFI,RMS_IFI_TMP
    50 01 D0 01F1 1349      MOVL    #1,R0
    04 01F4 1350      RET
    01F5 1351 :+++
    01F5 1352 : CLEAR_SYMBOL_NAME -- Clear SYMBOL_NAME prior to parsing SHOW/SYMBOL
    01F5 1353 :
    01F5 1354 : Inputs:
    01F5 1355 :
    01F5 1356 :     None.
    01F5 1357 :
    01F5 1358 : Outputs:
    01F5 1359 :
    01F5 1360 :     SYMBOL_NAME = 0
    01F5 1361 : ---
    01F5 1362
    0000 01F5 1363 CLEAR_SYMBOL_NAME:
    00000000'EF 0000 01F5 1364      .WORD 0
    7C 01 01F7 1365      CLRQ    SYMBOL_NAME
    50 01 D0 01FD 1366      MOVL    #1,R0
    04 0200 1367      RET

```

PARSE
V04-000

Parse tables for SDA commands
Action Routines for SET:SHOW RMS/DISPLAY

G 2

16-SEP-1984 01:36:01
5-SEP-1984 03:33:21

VAX/VMS Macro V04-00
[SDA.SRC]PARSE.MAR;1

Page 31
(12)

0201 1369
0201 1370

.END

POO
V04

PARSE
Symbol table

Parse tables for SDA commands

H 2

16-SEP-1984 01:36:01 VAX/VMS Macro V04-00
5-SEP-1984 03:33:21 [SDA.SRC]PARSE.MAR;1

Page 32
(12)

\$\$\$AST	= 00000000		
\$\$\$CNT	= 00000003		
\$\$\$FLG	= FFFFFFFF		
\$\$\$KEY	= 00000087		
\$\$\$KFG	= FFFFFFFF		
\$\$\$MOD	= 00000000		
\$\$\$TMP	= 00000398	R	05
\$\$KEYTAB	= 00000000	R	04
ADD2	00000000	R	06
ADD7FFE	000000A9	R	06
ADD8000	0000009E	R	06
ADDITION	00000510	R	03
ADDRESS	*****	X	06
ARGS	= 00000001		
ARITH_SHIFT	0000053A	R	03
ATM_OPT	00000250	R	03
CDT_BYADDR	*****	X	03
CDT_SPCFY	*****	X	03
CK EOS	00000046	R	03
CLEAR_SYMBOL_NAME	000001F5	R	06
CLOSE_LOG	*****	X	03
CLR_IF1_TMP	000001BC	R	06
COLLECT	000001EE	R	03
COLLECT_LOOP	000001F0	R	03
COLON	= 0000003A		
CONSTANT	0000059C	R	03
COPY_CMD	00000A1C	R	03
CSID	*****	X	03
CTLREG	000005D2	R	03
DECIMAL	000005BA	R	03
DEFINE	000009B2	R	03
DEFINE_KEY	00000122	RG	06
DEFINE_SYMBOL	*****	X	03
DEVICE	0000098E	R	03
DEVICE_OPTS	000009A2	R	03
DISPLAY	0000020E	R	03
DISPLAY_CRASH	*****	X	03
DISPLAY_DEVBYADDR	*****	X	03
DISPLAY_DEVICE	*****	X	03
DISPLAY_HELP	*****	X	03
DISPLAY_PFN	*****	X	03
DISPLAY_SPT	*****	X	03
DISPLAY_SPT_RANGE	*****	X	03
DIS_OPT	0000022C	R	03
DIS_OPT_EXIT	00000248	R	03
DIV2	0000001B	R	06
DIVISION	0000054E	R	03
END_SET_RMS	00000192	R	06
ESP	*****	X	06
EVALUATE	0000039E	R	03
EVAL_CONSTANT	0000003E	R	06
EVAL_CONTENTS	00000087	R	06
EVAL_DOT	0000007A	R	06
EVAL_EXP	000003AE	R	03
EVAL_QUAL	000003BE	R	03
EVAL_SETUP	00000048	R	06
EVAL_SYMBOL	00000062	R	06

EXAMINE	000003D0	R	03
EXAMINE_EXPRESSION	00000430	R	03
EXAM_INS	000004B0	R	03
EXAM_MEMORY	*****	X	03
EXAM_OK	000004A8	R	03
EXAM_QUALS	0000046C	R	03
EXAM_QUAL_LOOP	00000464	R	03
EXIT_CMD	000009E4	R	03
EXIT_COMMAND	*****	X	03
EXPOP	00000504	R	03
EXPRESSION	00000500	R	03
FACTOR	0000055A	R	03
FAILURE	0000005F	R	06
FILE_DESC	*****	X	03
FORMAT	*****	X	03
FORMAT_CMD	00000A2C	R	03
FORMAT_OPTS	00000A40	R	03
HELP_CMD	000009D0	R	03
HEX	000005BE	R	03
IF1	00000384	R	03
IF12	0000038C	R	03
INDEX	00000712	R	03
INDIRECT	0000058A	R	03
INDIRECT_CMD	0000007A	R	03
INDIRECT_COMMAND	*****	X	03
INPUT_BUFFER	*****	X	06
INPUT_LEN	*****	X	06
KEYTABLE	*****	X	06
KEY_ATTR	00000058	R	02
LIBSSIGNAL	*****	X	06
LOCK	000000D4	R	03
LOCKID	*****	X	03
LOCK_OPTS	000000EA	R	03
LOG_FILE	*****	X	03
MSG5_SUCCESS	*****	X	06
MULT2	00000009	R	06
MULTIPLICATION	00000544	R	03
NEG1	00000035	R	06
NEGATE	00000580	R	03
NXT_OPT	00000220	R	03
OCTAL	000005B6	R	03
OPEN_LOG	*****	X	03
OPEN_OUTPUT	*****	X	03
OPTSM_ALL	= 00000FFF		
OPTSM_ASB	= 00000040		
OPTSM_BAD	= 00000004		
OPTSM_BDB	= 00000010		
OPTSM_BDBSUM	= 00000020		
OPTSM_BLB	= 00008000		
OPTSM_BLBSUM	= 00010000		
OPTSM_CCB	= 00000080		
OPTSM_CHAN	= 00000400		
OPTSM_COND	= 00000100		
OPTSM_ESP	= 00000004		
OPTSM_FAB	= 00000400		
OPTSM_FCB	= 00000200		
OPTSM_FREE	= 00000001		

POO
V04

PARSE
Symbol table

Parse tables for SDA commands

I 2

16-SEP-1984 01:36:01 VAX/VMS Macro V04-00
5-SEP-1984 03:33:21 [SDA.SRC]PARSE.MAR;1

Page 33
(12)

OPTSM_FWA = 00100000
OPTSM_GBD = 00020000
OPTSM_GBDSUM = 00200000
OPTSM_GBH = 00040000
OPTSM_GLOBAL = 00000001
OPTSM_HEADER = 00000080
OPTSM_IDX = 00000008
OPTSM_IFB = 00000002
OPTSM_IMAGE = 00000001
OPTSM_INST = 00000020
OPTSM_IRB = 00000004
OPTSM_IRP = 00000002
OPTSM_ISP = 00000001
OPTSM_KSP = 00000002
OPTSM_LCK = 00000200
OPTSM_LENGTH = 00000010
OPTSM_LRP = 00000020
OPTSM_MODIFIED = 00000002
OPTSM_NAM = 00001000
OPTSM_NO = 00000001
OPTSM_NONPAGED = 00000004
OPTSM_NOSKIP = 00000200
OPTSM_PO = 00000001
OPTSM_PO_PPT = 00000800
OPTSM_P1 = 00000002
OPTSM_P1_PPT = 00001000
OPTSM_PAGED = 00000008
OPTSM_PCB = 00000010
OPTSM_PHD = 00000020
OPTSM_PPT = 00000002
OPTSM_PPT_LEN = 00002000
OPTSM_PPT_RNG = 00004000
OPTSM_PSL = 00000040
OPTSM_PST = 00000004
OPTSM_RAB = 00000800
OPTSM_RANGE = 00000008
OPTSM_REGS = 00000008
OPTSM_RJB = 00400000
OPTSM_RLB = 00004000
OPTSM_RMS = 00000080
OPTSM_RMSALL = FFFFFFFF
OPTSM_RMSD = 00000100
OPTSM_SELF = 00000001
OPTSM_SINGLEPFN = 00000010
OPTSM_SRP = 00000200
OPTSM_SSP = 00000008
OPTSM_SUMMARY = 00000040
OPTSM_SYSPROC = 00000040
OPTSM_SYSTEM = 00000004
OPTSM_TIME = 00000080
OPTSM_TRC = 00080000
OPTSM_TYPE = 00000100
OPTSM_USP = 00000010
OPTSM_WCB = 00000100
OPTSM_WHOLEPFN = 00000008
OPTSM_WSL = 00000001
OPTSM_XAB = 00002000

OPTIONS ***** X 03
OUTPUT_FILE ***** X 03
PAGE_OPTION 000008F8 R 03
PAGE_OPTS 000008E8 R 03
PAGE_RANGE 0000091C R 03
PAGE_TABLE 000008CC R 03
PARENS 00000594 R 03
PDT_BYADDR ***** X 03
PFN_DATA 00000864 R 03
PFN_OPTION 00000890 R 03
PFN_OPTS 00000886 R 03
PIX_WIDTH ***** X 06
POOL 0000072E R 03
POOL_RANGE 000007C0 R 03
PREV_BUFFER 00000000 R 02
PREV_COMMAND 00000050 R 02
PRINT_ANY_STACK ***** X 03
PRINT_HEADER ***** X 03
PRINT_STACKS ***** X 03
PROCESS 000005DA R 03
PROCESS1 000005EE R 03
PROCESS_SUMMARY ***** X 03
PROC_INDEX ***** X 06
PROC_NAME ***** X 03
PROC_PIX 000006B4 R 03
PROC_PPT 000006C0 R 03
RADIO 000005AA R 03
READ_CMD 000009F6 R 03
READ_QUALS 00000A0C R 03
READ_SYMFIL ***** X 03
RELOCATE_BASE ***** X 06
REPEAT_COMMAND 00000117 RG 06
REPEAT_KEY ***** X 06
REQMEM ***** X 06
RES 00000102 R 03
RMS 000006EC R 03
RMS_DIS_OPT ***** X 06
RMS_DIS_TMP ***** X 06
RMS_DIS_TMP1 ***** X 03
RMS_IFI ***** X 06
RMS_IFI_TMP ***** X 06
SAVE_COMMAND 000000D3 RG 06
SAVE_DUMP ***** X 03
SCS_CONNECT 0000015E R 03
SCS_PORTS 00000178 R 03
SCS_RDT 00000192 R 03
SDA_KEY 00000000 RG 04
SDA_STATE 00000000 RG 03
SEARCH 0000004E R 03
SEARCH_MEMORY ***** X 03
SEMI = 0000003B
SET 000001B6 R 03
SETUP_DIS_OPT 000001C8 R 06
SETUP_RMS_TMP 00000157 R 06
SET_LOG 000001CE R 03
SET_OUTPUT 000001DE R 03
SET_PROC 00000700 R 03

P00
V04

PARSE
Symbol table

Parse tables for SDA commands

J 2

16-SEP-1984 01:36:01 VAX/VMS Macro V04-00
5-SEP-1984 03:33:21 [SDA.SRC]PARSE.MAR;1

Page 34
(12)

POO
V04

```

SET_PROCESS          ***** X 03
SET_RMS              00000204 R 03
SHIFT2              00000028 R 06
SHOW                00000088 R 03
SHOW_ALL_LOCKS      ***** X 03
SHOW_ALL_RES        ***** X 03
SHOW_CLUSTER        ***** X 03
SHOW_CONNECTIONS    ***** X 03
SHOW_EXPR           ***** X 03
SHOW_ONE_LOCK       ***** X 03
SHOW_ONE_RES        ***** X 03
SHOW_POOL           ***** X 03
SHOW_POOL_RANGE     ***** X 03
SHOW_PORTS          ***** X 03
SHOW_PROCESS        ***** X 03
SHOW_RMS            0000012C R 03
SHOW_RMS_OPT        ***** X 03
SHOW_RSPID          ***** X 03
SHOW_SCS            ***** X 03
SHOW_SUMMARY        00000942 R 03
SHOW_SYMBOL         ***** X 03
SMG$ADD KEY DEF     ***** X 06
SMG$DEFINE KEY      ***** X 06
SMG$M_KEY_TERMINATE = 00000002
SNARF               000001FA R 03
STACK               000007E6 R 03
STACK_RANGE         0000083E R 03
START               00000000 R 03
STORE               00000050 R 06
STORE_IF1_TMP       000001AE R 06
STORE_PROC_INDEX    000000BF R 06
STORE_RELOCATION      000000B4 R 06
STORE_STAR          000001D4 R 06
STORE_TMP1_OPT      00000169 R 06
STRUCTURE           ***** X 03
SUB2                00000012 R 06
SUBTRACTION         0000051A R 03
SYM                 00000976 R 03
SYMBOL              0000095A R 03
SYMBOL_DESC         ***** X 03
SYMBOL_NAME         ***** X 03
SYMBOL_OPTS         00000982 R 03
SYMBOL_QUALS        00000960 R 03
SYMBOL_VALUE        ***** X 06
SYSREG              000005CA R 03
TERM                00000524 R 03
TPASL_NUMBER        = 0000001C
TPASL_STRINGCNT     = 00000008
TPASL_STRINGPTR     = 0000000C
TPASL_TOKENCNT      = 00000010
TPAS_ALPHA          = 000001EE
TPAS_ANY            = 000001ED
TPAS_BLANK          = 000001F2
TPAS_DECIMAL        = 000001F3
TPAS_DIGIT          = 000001EF
TPAS_EOS            = 000001F7
TPAS_EXIT           = FFFFFFFF

```

```

TPAS_FAIL           = FFFFFFFE
TPAS_FILESPEC       = 000001EA
TPAS_HEX            = 000001F5
TPAS_IDENT          = 000001EC
TPAS_KEYWORD        = 00000100
TPAS_LAMBDA         = 000001F6
TPAS_MAXKEY         = 000000DC
TPAS_OCTAL          = 000001F4
TPAS_STRING         = 000001F0
TPAS_SUBXPR         = 000001F8
TPAS_SYMBOL         = 000001F1
TPAS_UIC            = 000001EB
VALIDATE            000004C2 R 03
VALIDATE_QUEUE      ***** X 03
VALIDATE_Q_OK       000004F8 R 03
VAL_QUEUE           000004C6 R 03
VAL_QUEUE_OPTS      000004D2 R 03
VAL_QUE_QUAL        000004DC R 03
VAL_SET_MAX         ***** X 03
VAXCLUSTER          00000136 R 03

```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SDADATA	0000005C (92.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
LIB\$STATES	00000A5E (2654.)	03 (3.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE
LIB\$KEY0\$	00000110 (272.)	04 (4.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC WORD
LIB\$KEY1\$	0000039E (926.)	05 (5.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC WORD
PARSE	00000201 (513.)	06 (6.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:02.56
Command processing	132	00:00:00.50	00:00:09.59
Pass 1	1216	00:01:05.52	00:04:07.53
Symbol table sort	0	00:00:00.61	00:00:02.04
Pass 2	260	00:00:14.41	00:00:53.72
Symbol table output	37	00:00:00.17	00:00:00.38
Psect synopsis output	2	00:00:00.04	00:00:00.39
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1684	00:01:21.32	00:05:16.22

The working set limit was 3000 pages.
519055 bytes (1014 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 596 non-local and 3 local symbols.
1370 source lines were read in Pass 1, producing 69 object records in Pass 2.
32 pages of virtual memory were used to define 26 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1	4
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	14

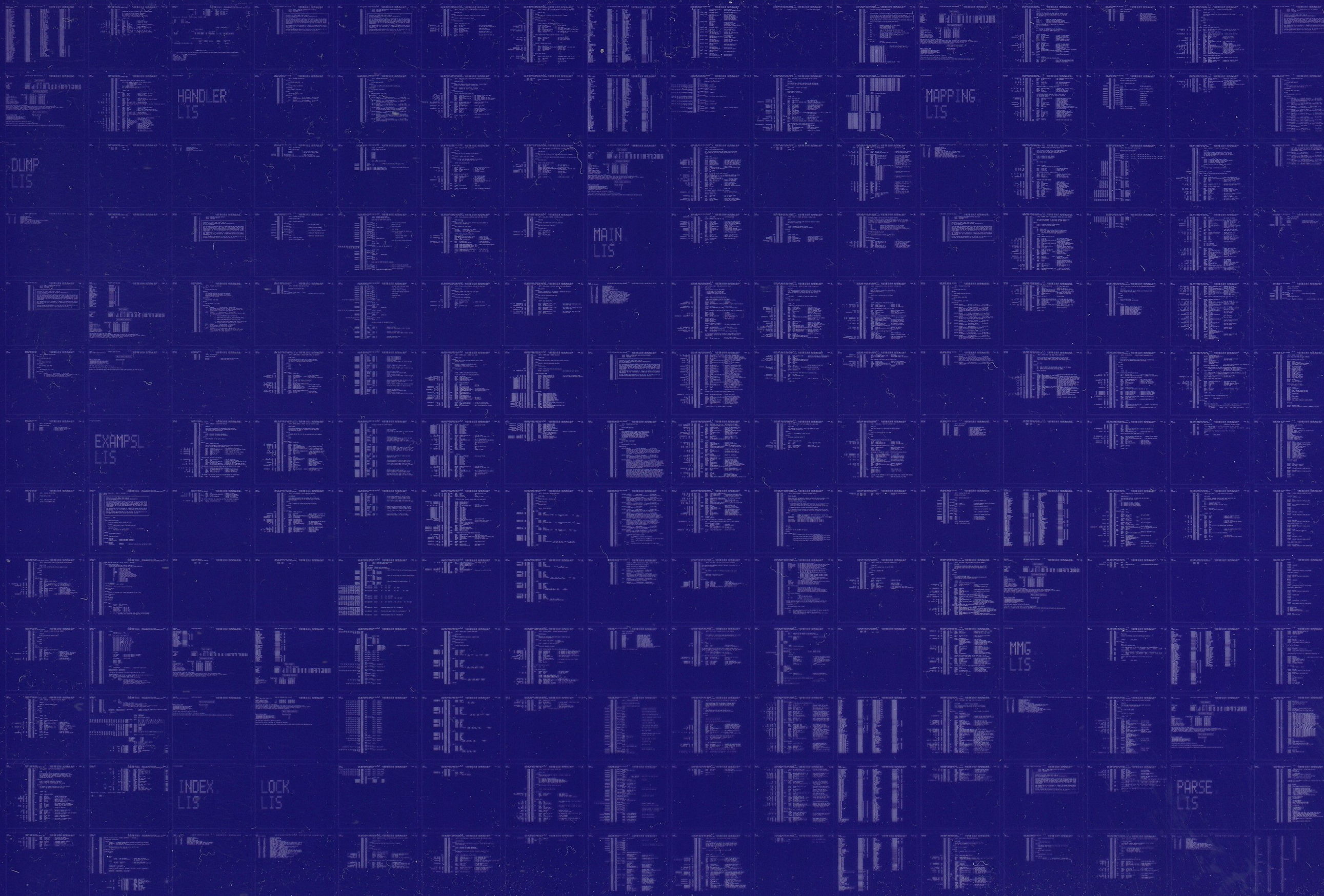
753 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:PARSE/OBJ=OBJ\$:PARSE MSRC\$:PARSE/UPDATE=(ENH\$:PARSE)+EXECML\$/LIB+LIB\$:SDALIB/LIB

0352 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0353

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY